Chapter 5

2-D Finite Difference

5.1 2-D Poisson's equation

Solving Laplace's ($f = 0$) or Poisson's equation in 2-D:

$$U_{xx} + U_{yy} = f \tag{5.1}$$

We discretise in x and y-directions:

We discretise in x-direction with grid spacing: $\Delta x = a/(m+1)$ and in y-direction with grid spacing: $\Delta y = b/(n+1)$, and let $x_k = k\Delta x$ where $0 \leq k \leq m+1$ and $y_j = j\Delta y$ where $0 \leq j \leq n+1$. We let $U_{kj} = U(x_k, y_j)$ and $f_{kj} = f(x_k, y_j)$ We are solving equation (5.1) using Dirichlet boundary conditions:

$$
\begin{aligned}
U(0,y) &= U_{0,j} = g_{0,j}(y_j) \\
U(a,y) &= U_{m+1,j} = g_{m+1,j}(y_j) \\
U(x,0) &= U_{k,0} = g_{k,0}(x_k) \\
U(x,b) &= U_{k,n+1} = g_{k,n+1}(x_k)
\end{aligned}
$$

Using central difference approximations for U_{xx} and U_{yy} then the finite difference approximations for equation (5.1) are given by:

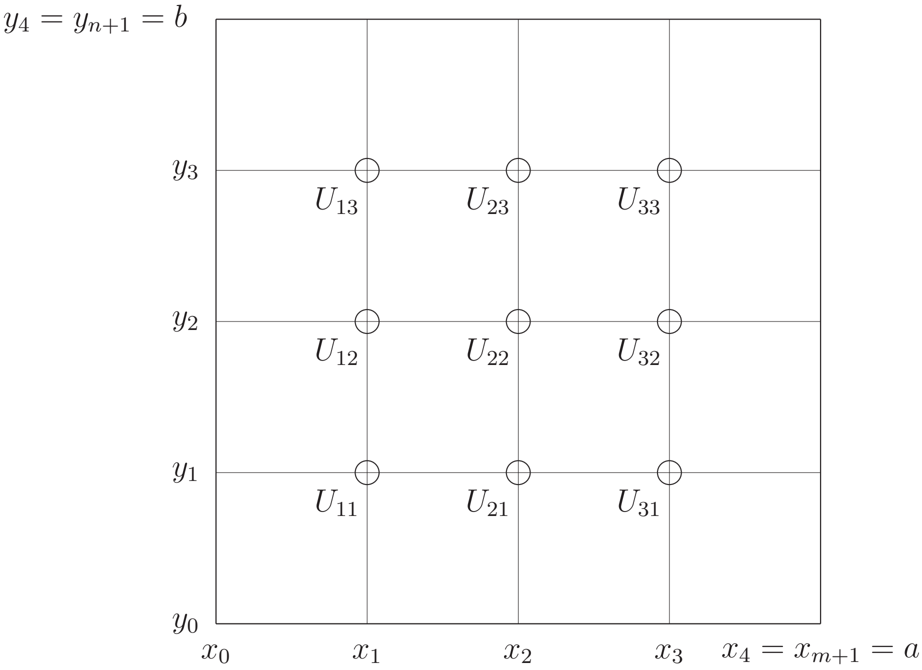
$$
\begin{aligned}
\frac{\partial^2 U}{\partial x^2} = U_{xx}(x_k, y_j) &= \frac{U_{k+1,j} - 2U_{k,j} + U_{k-1,j}}{\Delta x^2}, \\
\frac{\partial^2 U}{\partial y^2} = U_{yy}(x_k, y_j) &= \frac{U_{k,j+1} - 2U_{k,j} + U_{k,j-1}}{\Delta y^2}.
\end{aligned}
$$

Our discretised PDE (equation 5.1) becomes (if $\Delta x = \Delta y = h$):

$$
\begin{aligned}
\frac{U_{k+1,j} - 2U_{k,j} + U_{k-1,j}}{h^2} + \frac{U_{k,j+1} - 2U_{k,j} + U_{k,j-1}}{h^2} &= f_{k,j}, \\
\text{or } U_{k+1,j} + U_{k-1,j} - 4U_{k,j} + U_{k,j+1} + U_{k,j-1} &= h^2 f_{k,j}. \qquad (5.2)
\end{aligned}
$$

Since we have Dirichlet boundary conditions: the outer boundaries of the region we are solving for are known: $U_{0,j}, U_{m+1,j}, U_{k,0}, U_{k,n+1}$, and we need to find the interior values: $U_{k,j}$ for $1 \leq k \leq m$ and $1 \leq j \leq n$.

For example: $m = 3$ and $n = 3$:



Thus we need to solve for the interior values marked with a circle above as the boundary values are already given. We let the vector of interior values we are solving for be defined as:

$$\vec{U} = \begin{pmatrix} U_{11} \\ U_{12} \\ U_{13} \\ U_{21} \\ U_{22} \\ U_{23} \\ U_{31} \\ U_{32} \\ U_{33} \end{pmatrix}, \text{ vector of interior values we are solving for.}$$

Thus the matrix system for $\vec{U}$ using equation (5.2):

$U_{k+1,j} + U_{k-1,j} - 4U_{k,j} + U_{k,j+1} + U_{k,j-1} = h^2 f_{k,j}$ becomes:

$$
\begin{array}{c}
k=1 \\ j=1 \\ k=1 \\ j=2 \\ k=1 \\ j=3 \\ k=1 \\ j=1 \\ k=1 \\ j=2 \\ k=2 \\ j=3 \\ k=2 \\ j=1 \\ k=3 \\ j=2 \\ k=3 \\ j=3
\end{array}
\begin{pmatrix}
-4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
U_{k,j} & U_{k,j+1} & & U_{k+1,j} & & & & & \\
1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
U_{k,j-1} & U_{k,j} & U_{k,j+1} & & U_{k+1,j} & & & & \\
0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\
 & U_{k,j-1} & U_{k,j} & & & U_{k+1,j} & & & \\
1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\
U_{k-1,j} & & & U_{k,j} & U_{k,j+1} & & U_{k+1,j} & & \\
0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\
 & U_{k-1,j} & & U_{k,j-1} & U_{k,j} & U_{k,j+1} & & U_{k+1,j} & \\
0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\
 & & U_{k-1,j} & & U_{k,j-1} & U_{k,j} & & & U_{k+1,j} \\
0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\
 & & & U_{k-1,j} & & & U_{k,j} & U_{k,j+1} & \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\
 & & & & U_{k-1,j} & & U_{k,j-1} & U_{k,j} & U_{k,j+1} \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 \\
 & & & & & U_{k-1,j} & & U_{k,j-1} & U_{k,j}
\end{pmatrix}
\begin{pmatrix}
U_{11} \\ U_{12} \\ U_{13} \\ U_{21} \\ U_{22} \\ U_{23} \\ U_{31} \\ U_{32} \\ U_{33}
\end{pmatrix}
$$

$$
+ \begin{pmatrix}
U_{10} + U_{01} \\
U_{k,j-1} + U_{k-1,j} \\
U_{02} \\
U_{k-1,j} \\
U_{14} + U_{03} \\
U_{k,j+1} + U_{k-1,j} \\
U_{20} \\
U_{k,j-1} \\
0 \\
\\
U_{24} \\
U_{k,j+1} \\
U_{30} + U_{41} \\
U_{k,j-1} + U_{k+1,j} \\
U_{42} \\
U_{k+1,j} \\
U_{34} + U_{43} \\
U_{k,j+1} + U_{k+1,j}
\end{pmatrix}
= h^2 \begin{pmatrix}
f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33}
\end{pmatrix}
$$

ie:

$$\begin{pmatrix} -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 \end{pmatrix} \begin{pmatrix} U_{11} \\ U_{12} \\ U_{13} \\ U_{21} \\ U_{22} \\ U_{23} \\ U_{31} \\ U_{32} \\ U_{33} \end{pmatrix}$$

$$\begin{aligned} &= \begin{pmatrix} h^2 f_{11} - g_{10} - g_{01} \\ h^2 f_{12} - g_{02} \\ h^2 f_{13} - g_{14} - g_{03} \\ h^2 f_{21} - g_{20} \\ h^2 f_{22} \\ h^2 f_{23} - g_{24} \\ h^2 f_{31} - g_{30} - g_{41} \\ h^2 f_{32} - g_{42} \\ h^2 f_{33} - g_{34} - g_{43} \end{pmatrix}, \\ \text{or} \qquad A\vec{U} &= \vec{f}. \end{aligned} \tag{5.3}$$

If $m \times n \leq 100$ then direct matrix elimination methods can be used. Otherwise iterative methods such as Jacobi, Gauss-Seidel, relaxation methods should be used to solve for $\vec{U}$, as discussed in previous chapter 4. Because A is sparse and diagonally dominant, iterative solutions are ideal here.

However we see in the next section 5.2.1 that when we solve 2-D parabolic equations (2-D heat or diffusion equations) that the numerical solution requires 'tweaking' since the matrix A is no longer tridiagonal for 2-D as it was for 1-D.

5.2 2-D Heat (or Diffusion) Equation

We consider the 2-D heat equation:

$$U_t = \beta(U_{xx} + U_{yy}) \quad \text{for} \quad 0 \leq x \leq a, \quad 0 \leq y \leq b \quad \text{and} \quad 0 \leq t \leq T.$$

with initial conditions: $U(0, x, y) = f(x, y)$ and boundary conditions: $U(t, x, y) = g(t, x, y)$ for (x, y) on boundary.

- If we use explicit forward Euler scheme as we did for the 1-D heat equation the stability criteria is even stricter than for 1-D:

$$\Delta t \leq \frac{\Delta x^2 + \Delta y^2}{8\beta}$$

$\rightarrow$ this is less attractive because the time step is much smaller.

- if we use backward Euler or the Crank-Nicolson methods they are no longer as attractive because the matrix systems to be solved are much larger and *no longer tridiagonal.*

- We will look at an alternative finite difference method specifically tailored to the 2-D heat equation: the alternating direction implicit (ADI) method.

5.2.1 Alternating Direct/Implicit method for the 2-D heat equation

$$U_t = \beta(U_{xx} + U_{yy})$$

We let $\Delta t = T/m,\ \Delta x = a/(n+1),\ \Delta y = b/(p+1)\ t_k = k\Delta t,\ 0 \leq k \leq m,\ x_i = i\Delta x, 0 \leq i \leq n+1,\ y_j = j\Delta y,\ 0 \leq j \leq p+1$, and:

$$U(t_k, x_i, y_j) = U_{ij}^k$$

The time derivative, U_t, is approximated using a leap-frog step in time about the mid-point $t_{k+1/2}$, where $t_{k+1/2} = (t_k + t_{k+1})/2$ using a time step of $\Delta t/2$:

$$\frac{\partial U_{ij}^{k+1/2}}{\partial t} = \frac{U_{ij}^{k+1} - U_{ij}^{k-1}}{\Delta t} \qquad \text{at time } t_{k+1/2}.$$

The spatial derivatives, U_{xx} and U_{yy}, are approximated using central differences:

$$\frac{\partial^2 U_{ij}^k}{\partial x^2} = \underbrace{\frac{U_{i+1,j}^k - 2U_{ij}^k + U_{i-1,j}^k}{\Delta x^2}}_{\text{EARLY STEP}} \qquad \text{at time } t_k,$$

$$\frac{\partial^2 U_{ij}^{k+1}}{\partial y^2} = \frac{U_{i,j+1}^{k+1} - 2U_{ij}^{k+1} + U_{i,j-1}^{k+1}}{\Delta y^2} \qquad \text{at time } t_{k+1}.$$

This introduces an 'early' bias which evaluates $U_{xx}(t_k)$ at an earlier time than $U_{yy}(t_{k+1})$. This bias is compensated by also evaluating $U_{xx}(t_{k+2})$ at t_{k+2}; $U_{yy}(t_{k+1})$ again at t_{k+1} and $U_t(t_{k+3/2})$ at mid-point between t_{k+1} and t_{k+2}:

$$\begin{aligned}
\frac{\partial U_{ij}^{k+3/2}}{\partial t} &= \frac{U_{ij}^{k+2} - U_{ij}^{k+1}}{\Delta t}, \\
\frac{\partial^2 U_{ij}^{k+2}}{\partial x^2} &= \underbrace{\frac{U_{i+1,j}^{k+2} - 2U_{ij}^{k+2} + U_{i-1,j}^{k+2}}{\Delta x^2}}_{\text{LATE STEP}} \\
\text{and} \qquad & \frac{\partial^2 U_{ij}^{k+1}}{\partial y^2} \text{ as before.}
\end{aligned}$$

The boundary conditions specify U_{oj}^k, $U_{m+1,j}^k$, U_{io}^k, $U_{i,p+1}^k$, and initial conditions specify U_{ij}^0. So we are solving for $1 \le k \le m$, $1 \le i \le m$, $1 \le j \le p$. ie. for each interior point at time t_k.
We solve the problem for both time steps t_{k+1} and t_{k+2} at the same time using early and late definitions. If we let $s_x = \beta \Delta t / \Delta x^2$, $s_y = \beta \Delta t / \Delta y^2$ then $U_t = \beta(U_{xx} + U_{yy})$ becomes:
Early step:

$$U_{ij}^{k+1}(1 + 2s_y) - s_y[U_{i,j+1}^{k+1} + U_{i,j-1}^{k+1}] = U_{ij}^k(1 - 2s_x) + s_x[U_{i+1,j}^k + U_{i-1,j}^k]$$

This is solved first for i^{th} row of U^{k+1} matrix, for $1 \le i \le n$.

Late step:

$$U_{ij}^{k+2}(1 + 2s_x) - s_x[U_{i+1,j}^{k+2} + U_{i-1,j}^{k+2}] = U_{ij}^{k+1}(1 - 2s_y) + s_y[U_{i,j+1}^{k+1} + U_{i,j-1}^{k+1}]$$

This is solved for j^{th} column of U^{k+2} matrix, for $1 \le j \le p$.
These are solved using LU decomposition. (For more details see Schilling and Harris, p. 445).

5.3 Cylindrical and spherical polar co-ordinates

- Spherical and cylindrical symmetry in problems are often exploited to reduce 2-D $\to$ 1-D or 3-D $\to$ 1-D.

3-D Cylindrical Co-ordinates

$x = r \cos\theta$
$y = r \sin\theta$

$z = z$
$\Rightarrow r = \sqrt{x^2 + y^2}$
$\theta = \arctan(\frac{y}{x})$

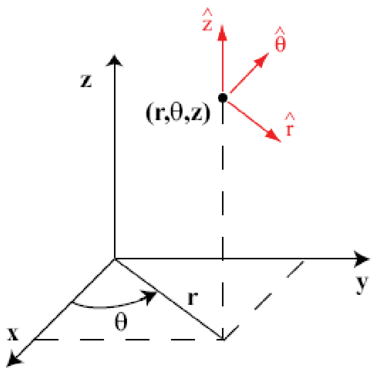


Figure 5.1: 3-D Cylindrical Co-ordinates

3-D Spherical Polar Co-ordinates

$x = r \sin\theta \cos\phi$
$y = r \sin\theta \sin\phi$
$z = r \cos\theta$

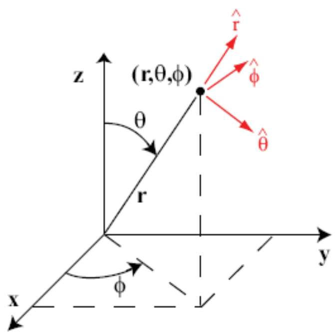


Figure 5.2: 3-D Spherical Polar Co-ordinates

2-D Polar Co-ordinates

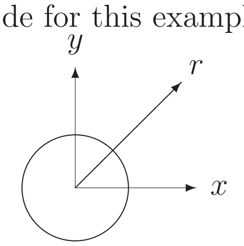
$x = r \cos\phi$
$y = r \sin\phi$

$\Rightarrow r = \sqrt{x^2 + y^2}$
$\phi = \arctan(\frac{y}{x})$.

$$\frac{\partial}{\partial x} = \frac{\partial r}{\partial x}\frac{\partial}{\partial r} + \frac{\partial \phi}{\partial x}\frac{\partial}{\partial \phi} = \cos\phi\frac{\partial}{\partial r} - \frac{\sin\phi}{r}\frac{\partial}{\partial \phi}$$
$$\frac{\partial}{\partial y} = \frac{\partial r}{\partial y}\frac{\partial}{\partial r} + \frac{\partial \phi}{\partial y}\frac{\partial}{\partial \phi} = \sin\phi\frac{\partial}{\partial r} + \frac{\cos\phi}{r}\frac{\partial}{\partial \phi}$$

5.3.1 Example: Temperature around a nuclear waste rod

The matlab code for this example is **NuclearWaste.m**.



Nuclear rod buried in ground

We consider the temperature increase due to storage of nuclear rods which release heat due to radioactive decay:

$$\underbrace{\frac{1}{\kappa}\frac{\partial T}{\partial t}(r,t) - \nabla^2 T(r,t)}_{\text{2 -D heat equation}} = \underbrace{S(r,t)}_{\text{source term}}$$

where the source term due to the radioactive decay of rod is given by:

$$S(r,t) = \begin{cases} T_{rod}e^{-t/\tau_0}/a^2 & \text{for } r \leq a \\ 0 & \text{elsewhere.} \end{cases}$$

where $a = 25$cm, $\kappa = 2 \times 10^7$cm^2/year, $T_{rod} = 1K$, $\tau_0 = 100$years, $r_c = 100$cm, $T_E = 300K$, $0 < r < 100$cm and $0 < t < 100$years. Initially $T(r, t = 0) = 300K$.

Because the problem has circular symmetry (ie. no ϕ dependence) $\Rightarrow$ 2-D problem in (x, y) reduced to 1-D problem in r. $\nabla^2 T = T_{xx} + T_{yy}$ is 2-D in Cartesian co-ordinates. However if we choose to use polar co-ordinates then the temperature, $T(r, t)$ is a function of r and t only because the rod circularly symmetric and there is no ϕ dependence. This reduces the original

2-D problem to 1-D!
How do we evaluate $\bigtriangledown^2 T$ in polar co-ordinates?

$$\begin{aligned}
T_{xx} &= \frac{\partial^2 T}{\partial x^2} = (\cos\phi\frac{\partial}{\partial r} - \frac{\sin\phi}{r}\frac{\partial}{\partial\phi})(\cos\phi\frac{\partial T}{\partial r} - \frac{\sin\phi}{r}\frac{\partial T}{\partial\phi}) \\
&= \cos^2\phi T_{rr} - \frac{2\sin\phi\cos\phi}{r}T_{r\phi} + \frac{\sin^2\phi}{r}T_r + \frac{2\cos\phi\sin\phi}{r}T_\phi + \frac{\sin^2\phi}{r}T_{\phi\phi}
\end{aligned}$$

$$\begin{aligned}
T_{yy} &= (\sin\phi\frac{\partial}{\partial r} + \frac{\cos\phi}{r}\frac{\partial}{\partial\phi})(\sin\phi\frac{\partial T}{\partial r} + \frac{\cos\phi}{r}\frac{\partial T}{\partial\phi}) \\
&= \sin^2\phi T_{rr} + \frac{2\cos\phi\sin\theta}{r}T_{r\phi} - \frac{2\cos\phi\sin\phi}{r^2}T_\phi + \frac{\cos^2\phi}{r}T_r + \frac{\cos^2\phi}{r^2}T_{\phi\phi}
\end{aligned}$$

$$\text{and } T_{xx} + T_{yy} = T_{rr} + \frac{1}{r}T_r + \frac{1}{r}^2 T_{\phi\phi}$$

using $\cos^2\phi + \sin^2\phi = 1$.

Since the temperature $T(r,t)$ has no ϕ dependence then $T_{\phi\phi} = 0$ and we are solving the 1-D heat equation in polar co-ordinates:

$$\frac{1}{K}\frac{\partial T}{\partial t} - \frac{\partial^2 T}{\partial r^2} - \frac{1}{r}\frac{\partial T}{\partial r} = S(r,t)$$

We know that in the steady state solution eventually the nuclear rod is no longer radioactive and stops releasing heat: $S(r,t) \to 0$ as $t \to \infty$, and further enough away from the rod the temperature equals the environment temperature, $T(r = r_c, t) = 300K$. So the solution should approach the environmental temperature $T(r,t) = 300K$ once rod has finished radioactive decaying.
We use finite differences to solve:

$$\frac{1}{K}\frac{\partial T}{\partial t} - \frac{\partial^2 T}{\partial r^2} - \frac{1}{r}\frac{\partial T}{\partial r} = S(r,t) \tag{5.4}$$

We observe that there is a singularity at $r = 0$ in the above equation where special care needs to be taken so that the numerical solution is stable.

Initial conditions $T(r,0) = 300K$.
Neumann boundary conditions at $r = 0$ (temperature cannot flow into $r = 0$ region)

$$\frac{\partial T}{\partial r}(r = 0, t) = 0$$

Dirichlet boundary conditions at $r = r_c$

$$T(r = r_c, t) = 300K$$

Again we discretise space and time: $\Delta r = r_c/(n+1)$, $\Delta t = T_f/m$, $r_j = j\Delta r$, $0 \le j \le n+1$, $t_k = k\Delta t$, $0 \le k \le m$, $T(r_j, t_k) = T_j^k$, and $S(r_j, t_k) = S_j^k$.

Discrete Neumann boundary conditions at $r = 0$ become:

$$\frac{\partial T_j^k}{\partial t}(r = 0, t) = \frac{\partial T_0^k}{\partial t} = 0 \approx \frac{T_1^k - T_0^k}{\Delta t} \Rightarrow T_0^k \approx T_1^k$$

Discrete Dirichlet boundary conditions at $r = r_c$ become:

$$T_j^k(r = r_c, t) = T_{n+1}^k = 300$$

We will use the backward Euler method (*implicit*) to solve the PDE. This means evaluating the spatial derivatives in r at the future time step t_{k+1}:

$$\begin{aligned}
T_t(t_{k+1}, r_j) &= \frac{T_j^{k+1} - T_j^k}{\Delta t} \\
T_{rr}(t_{k+1}, r_j) &= \frac{T_{j+1}^{k+1} - 2T_j^{k+1} + T_{j-1}^{k+1}}{\Delta r^2} \text{ (centred difference at } t_{k+1}) \\
T_r(t_{k+1}, r_j) &= \frac{T_{j+1}^{k+1} - T_{j-1}^{k+1}}{2\Delta r} \text{ (leap-frog in space)}
\end{aligned}$$

Using $r_j = j\Delta r$ our discretised PDE 5.4 becomes:

$$\begin{aligned}
\frac{1}{\kappa}T_t - T_{rr} - \frac{1}{r}T_r &= S(r,t) \\
\underbrace{\frac{1}{\kappa\Delta t}[T_j^{k+1} - T_j^k]}_{T_t/\kappa} - \underbrace{[\frac{T_{j+1}^{k+1} - 2T_j^{k+1} + T_{j-1}^{k+1}}{\Delta r^2}]}_{T_{rr}} - \underbrace{\frac{1}{j\Delta r}[\frac{T_{j+1}^{k+1} - T_{j-1}^{k+1}}{2\Delta r}]}_{T_r/r} &= S_j^k
\end{aligned}$$

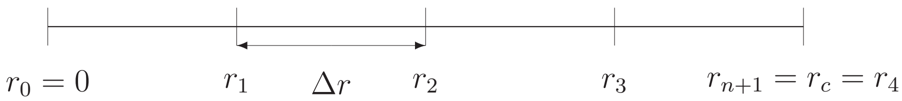
We let $s = \kappa\Delta t/\Delta r^2$ and we arrive at:

$$T_{j+1}^{k+1}[-s - \frac{s}{2j}] + T_{j-1}^{k+1}[-s + \frac{s}{2j}] + T_j^{k+1}[1 + 2s] = T_j^k + S_j^k\kappa\Delta t \qquad (5.5)$$

This is a tridiagonal matrix for $1 \le j \le n$.

Numerical solution of the 1-D heat equation in polar co-ordinates using the Backward Euler method

For $n = 3$:



The boundary conditions give $T_0^k \approx T_1^k$ using $\frac{\partial T}{\partial r}(r = 0, t) = 0)$ and $T_4^k = 300K$ using $(T(r = r_c, t) = 300)$ and the initial conditions are $T_j^0 = 300K$.

We solve equation 5.5 for T_1^k, T_2^k, T_3^k at each time step (t_k):

$$\begin{pmatrix} 1+2s & (-s-\frac{s}{2j}) & 0 \\ (-s+\frac{s}{2j}) & 1+2s & (-s-\frac{s}{2j}) \\ 0 & (-s+\frac{s}{2j}) & 1+2s \end{pmatrix} \begin{pmatrix} T_1^{k+1} \\ T_2^{k+1} \\ T_3^{k+1} \end{pmatrix} + \begin{pmatrix} (-s+\frac{s}{2j})T_0^{k+1} \\ 0 \\ (-s-\frac{s}{2j})T_4^{k+1} \end{pmatrix}$$
$$= \begin{pmatrix} T_1^k \\ T_2^k \\ T_3^k \end{pmatrix} + \kappa\Delta t \begin{pmatrix} S_1^k \\ S_2^k \\ S_3^k \end{pmatrix}$$

Using the boundary conditions: $T_0^{k+1} \approx T_1^{k+1}, T_4^{k+1} = 300K$

$$\begin{pmatrix} (1+s+\frac{s}{2j}) & (-s-\frac{s}{2j}) & 0 \\ (-s+\frac{s}{2j}) & (1+2s) & (-s-\frac{s}{2j}) \\ 0 & (-s+\frac{s}{2j}) & (1+2s) \end{pmatrix} \begin{pmatrix} T_1^{k+1} \\ T_2^{k+1} \\ T_3^{k+1} \end{pmatrix}$$
$$= \begin{pmatrix} T_1^k \\ T_2^k \\ T_3^k \end{pmatrix} + \kappa\Delta t \begin{pmatrix} S_1^k \\ S_2^k \\ S_3^k \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ (-s-\frac{s}{2j})T_4^{k+1} \end{pmatrix}$$
$$\Rightarrow \begin{pmatrix} (1+s+\frac{s}{2}) & (-s-\frac{s}{2}) & 0 \\ (-s+\frac{s}{4}) & (1+2s) & (-s-\frac{s}{4}) \\ 0 & (-s+\frac{s}{6}) & (1+2s) \end{pmatrix} \begin{pmatrix} T_1^{k+1} \\ T_2^{k+1} \\ T_3^{k+1} \end{pmatrix}$$
$$= \begin{pmatrix} T_1^k \\ T_2^k \\ T_3^k \end{pmatrix} + \kappa\Delta t \begin{pmatrix} S_1^k \\ S_2^k \\ S_3^k \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ (-s-\frac{s}{6})300 \end{pmatrix}$$

Or to simplify we are solving the following matrix equation for the vector of unknown temperatures $\vec{T}^{k+1}$:

$$A\vec{T}^{k+1} = \vec{T}^k + \kappa\Delta t\vec{S}^k + \vec{b}$$

The matlab code **NuclearWaste.m** can be used to check that solution for $T \to 300K$ as $t \to \infty$ (steady state approaches environment temperature, 300K).

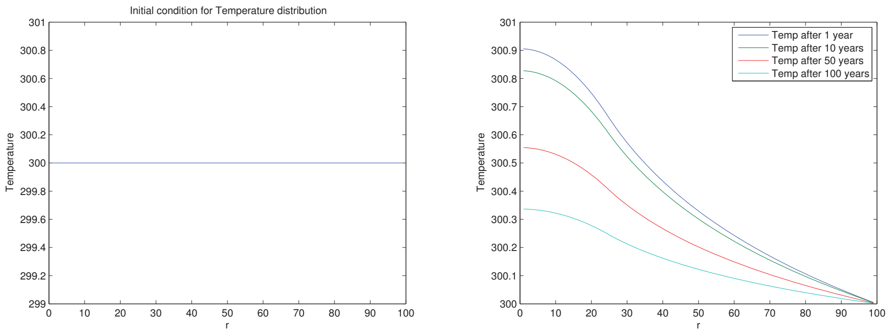


Figure 5.3: Initial conditions in (a) and matlab solution using Backward Euler method for temperature distribution near nuclear rod at different time intervals in (b)

Figure 5.3 shows the initial conditions and temperature distribution near the nuclear rod at different time intervals.